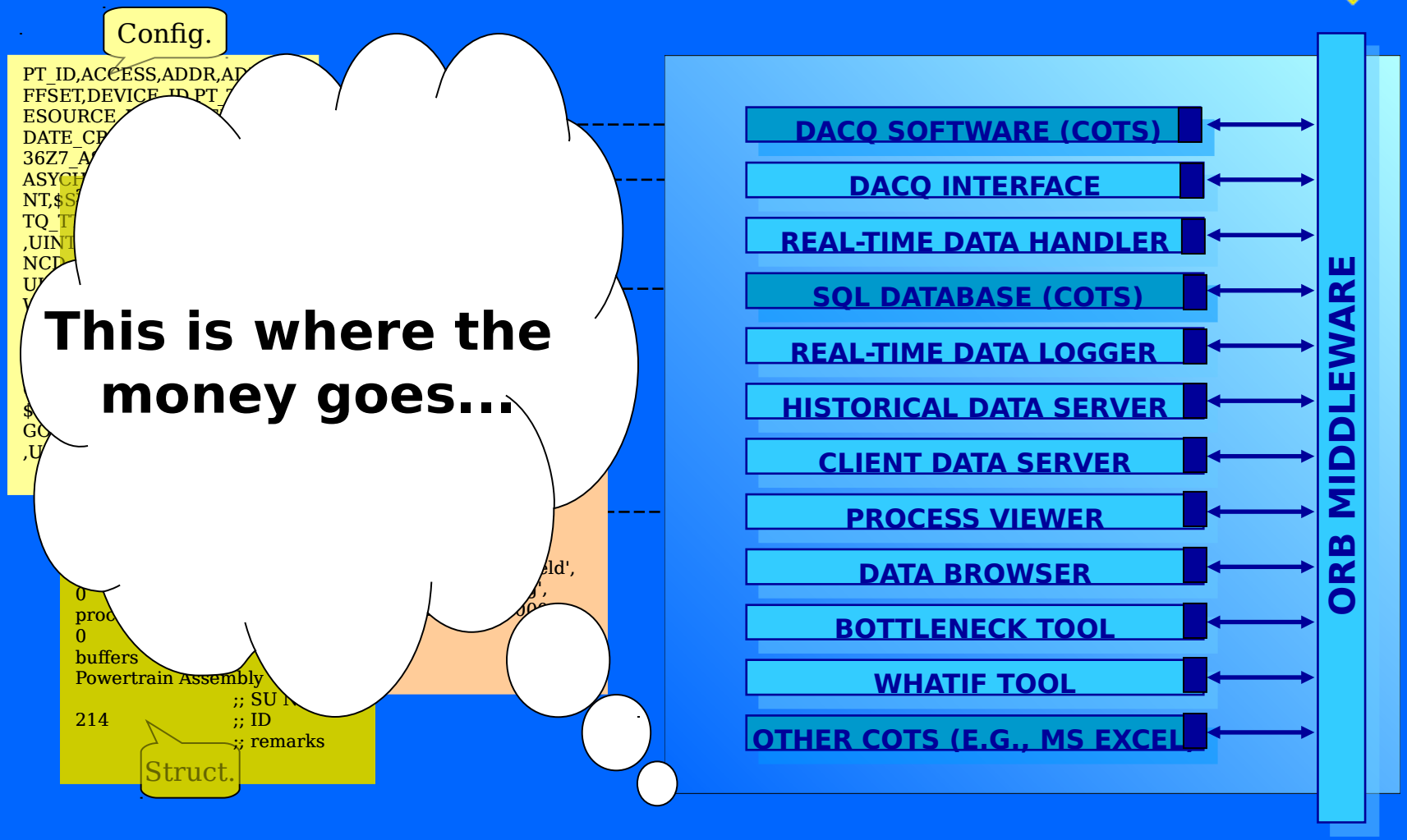
Model-Integrated Computing & EDCS Technologies

27-28 Apr 99

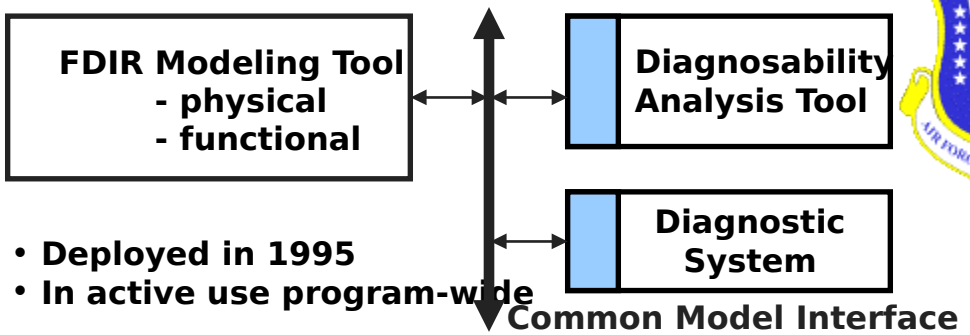
Roger J. Dziegiel, Jr
AFRL/IFTD
525 Brooks Rd
Rome, NY 13441-4505
(315)330-2185
dziegielr@rl.af.mil

Problems


Config.
PT_ID,ACCESS,ADDR,AD
FFSET,DEVICE_ID,PT_
ESOURCE_
DATE_CR
36Z7_A
ASYCH
NT,$S
TQ_T
,UINT
NCD
U
V
$
GC
,U
0
proc
0
buffers
Powertrain Assembly
214
;; SU
;; ID
;; remarks
Struct.
This is where the money goes...
DACQ SOFTWARE (COTS)
DACQ INTERFACE
REAL-TIME DATA HANDLER
SQL DATABASE (COTS)
REAL-TIME DATA LOGGER
HISTORICAL DATA SERVER
CLIENT DATA SERVER
PROCESS VIEWER
DATA BROWSER
BOTTLENECK TOOL
WHATIF TOOL
OTHER COTS (E.G., MS EXCEL)
ORB MIDDLEWARE

International Space Station

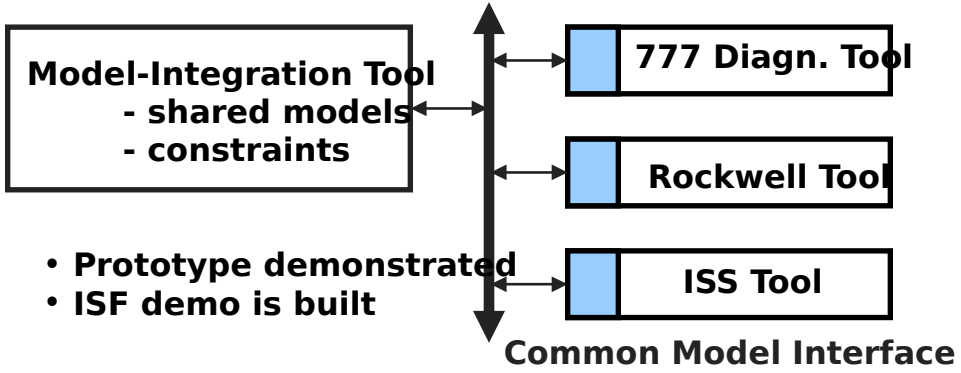
Fault Detection, Isolation and Recovery (FDIR)



- Deployed in 1995
- In active use program-wide

JSF/Boeing

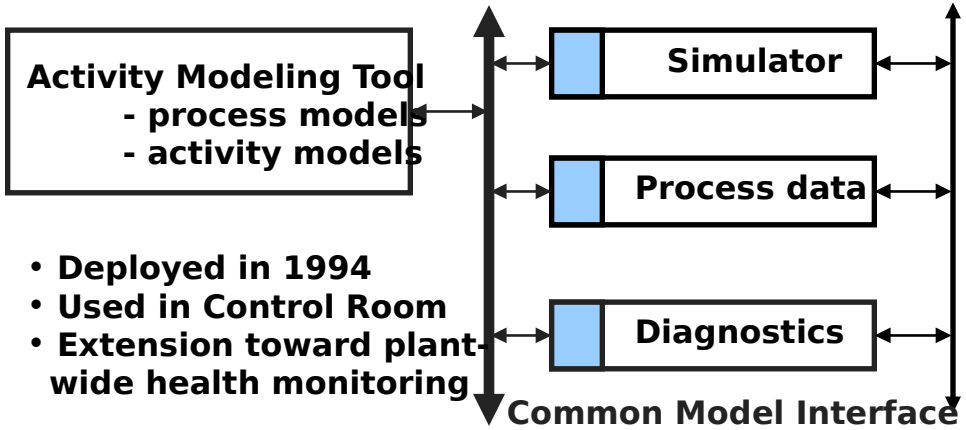
JSF Common Diagnostic System Testbench



- Prototype demonstrated
- ISF demo is built

DuPont Chemicals

On-line Problem Solving Environment for Plant Operation



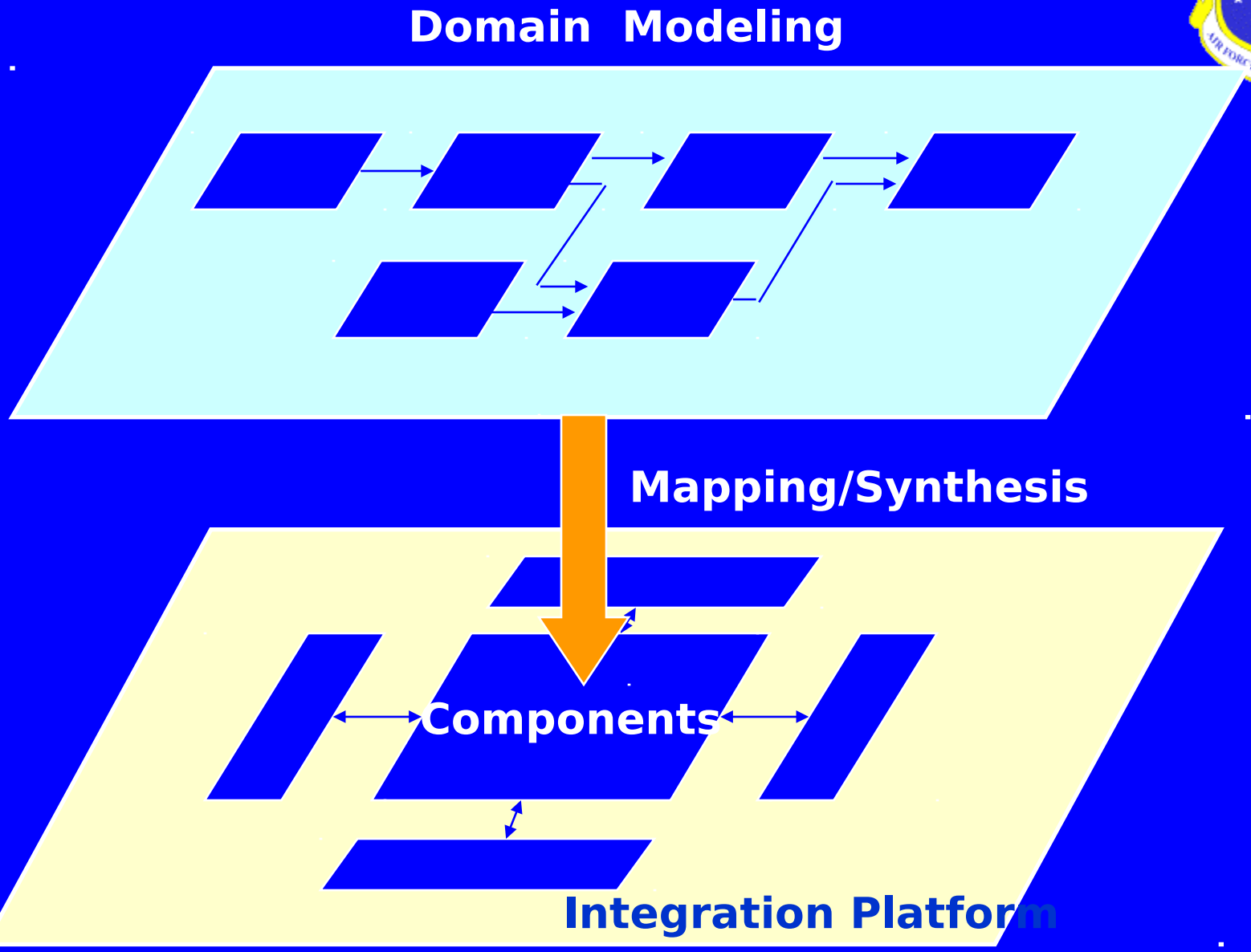
- Deployed in 1994
- Used in Control Room
- Extension toward plant-wide health monitoring

Model Integrated Computing (MIC)

Grew out of over 12 years of research on compute
based systems in aerospace, instrumentation,
manufacturing and robotics. Common challenges:

- "Software" and "environment" are inextricab
  combined
- Need for adaptability to changing environme
  and end-user needs
- Complex, heterogeneous applications
- Stringent reliability and dependability requir

Conceptual structure



General Considerations

- Domain space is dominated by a mature enginee discipline, the modeling paradigm is “not negot
- Scope of the modeling paradigm covers the full c
- The mapping between Domain Models and analys models used by Analysis Tools can be complex
- The mapping between Domain Models and Execu Models used for defining software applications can be complex and dynamic

Common Thread:

- Represent the information that directly or indirectly determines the structure of computations using **domain-specific modeling paradigms**
- Capture the relationship between **domain models**, **analysis models** and **executable models**
- Validate, analyze the domain models **using generic tools**
- **Synthesize** computations from the domain models

Model-Integrated Comput

Research Issues

- **Model Integration:**
  - How to facilitate building and verification of multiple view, domain-specific models?
- **Model Transformation:**
  - How to capture the mapping between Design Models Analysis Models and Executable Models?
  - How to synthesize (and re-synthesize) Executable Model (i.e. Applications)?
- **Executable Models**
  - How to create dynamically reconfigurable computation m
  - How to create run-time support for hardware reconfigura
- **Reusable architectural framework**

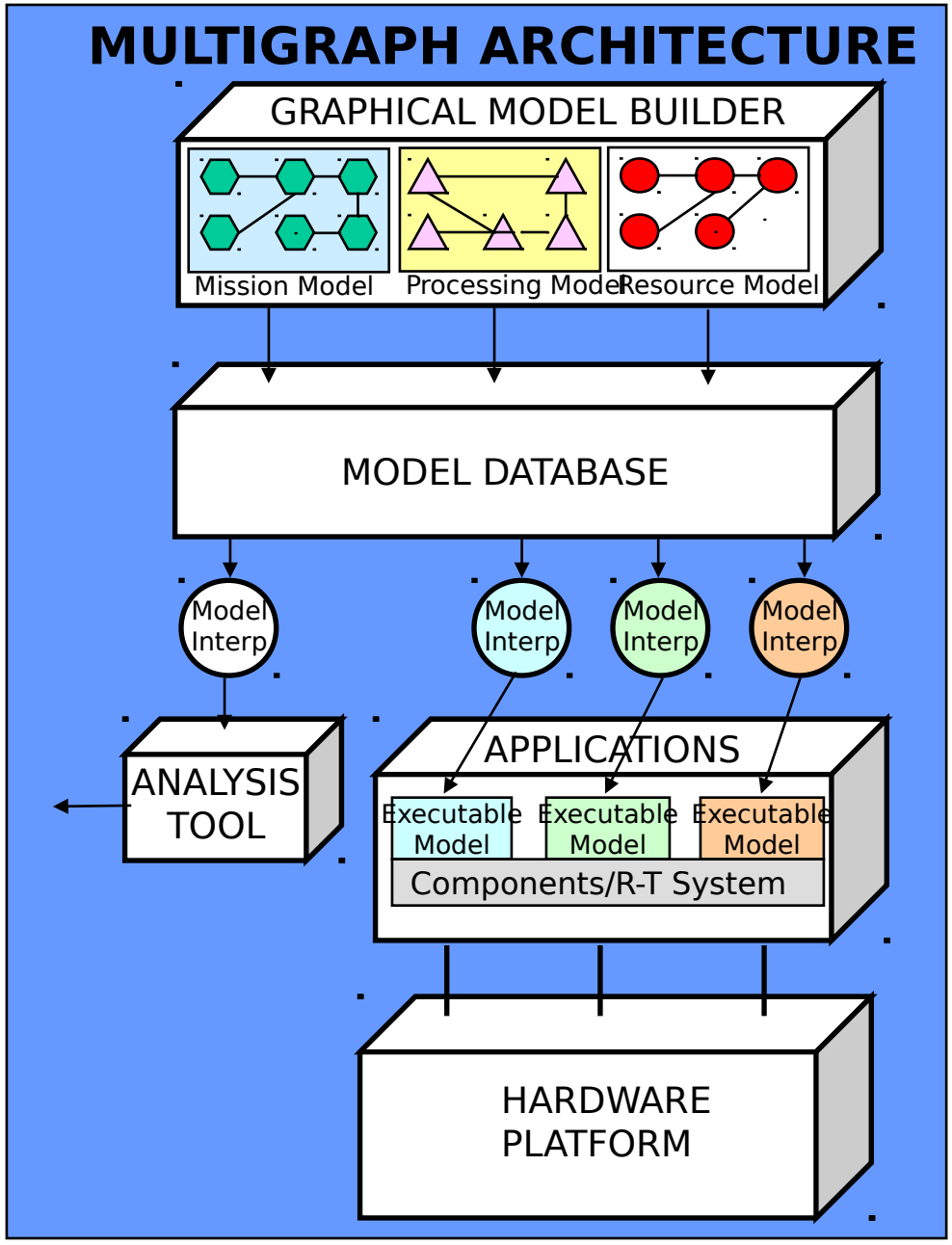
MultiGraph Architecture

Challenges:

separate generic and specific, make the technology affordable

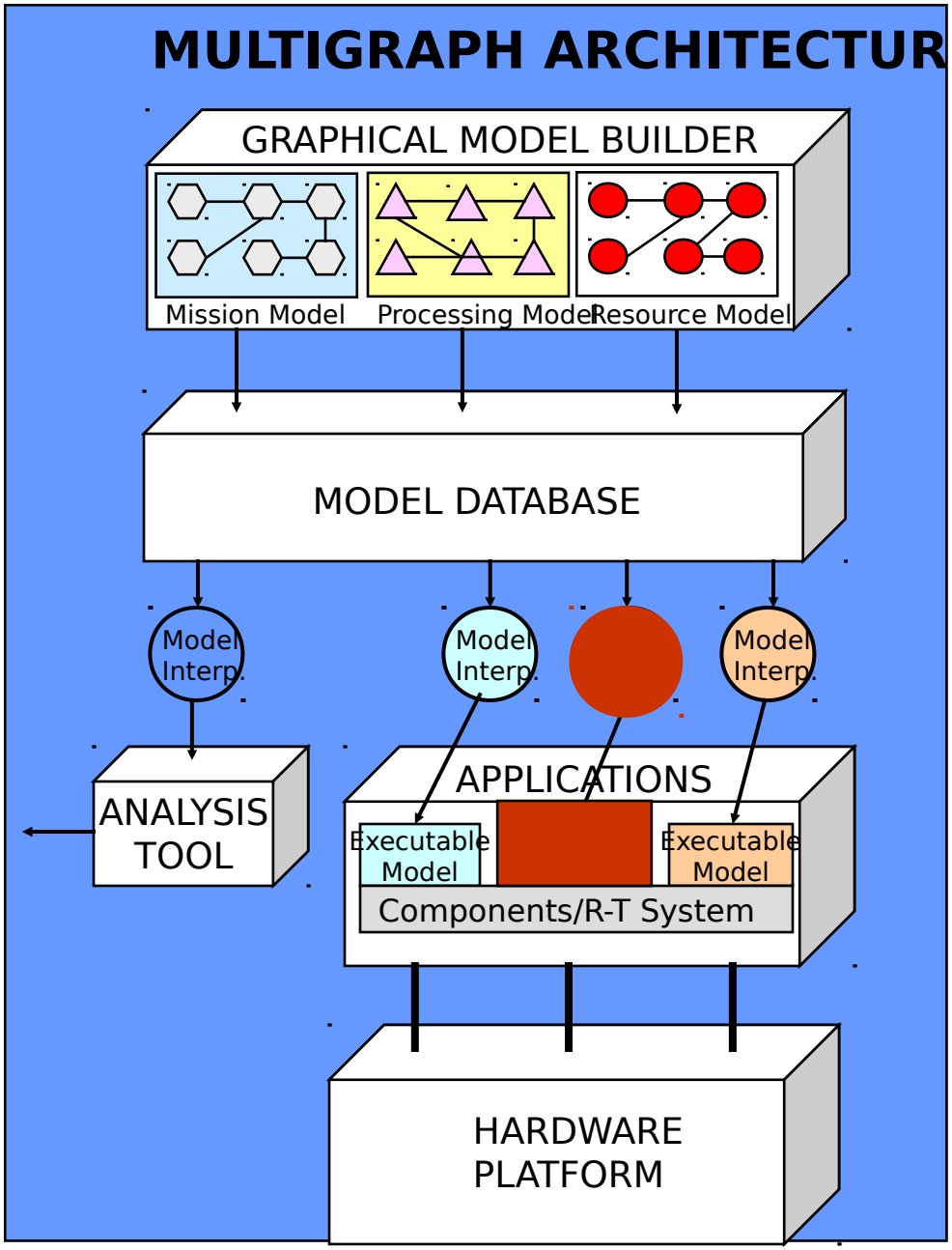
MGA:

Infrastructure for building domain specific Model-Integrated Program Synthesis Environments



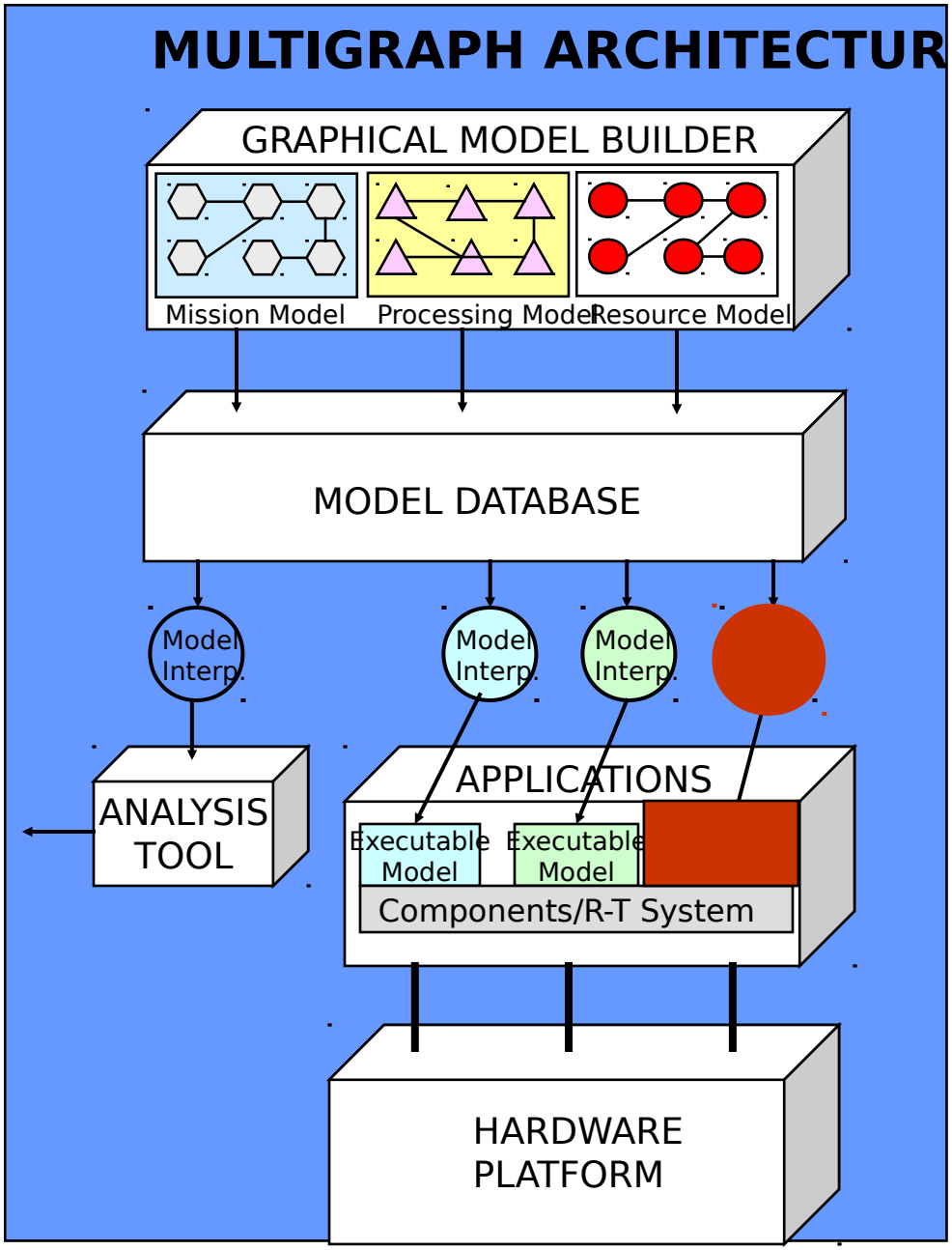
MIC APPROACH IN THE MG

- **MODEL INTEGRATION:** Multiple, "overlapping" views with maintained constraints
- **MODEL TRANSFORMATION:** Model Interpreters transl domain models into analy and executable models
- **APPLICATION INTEGRATION** run-time support for executable models



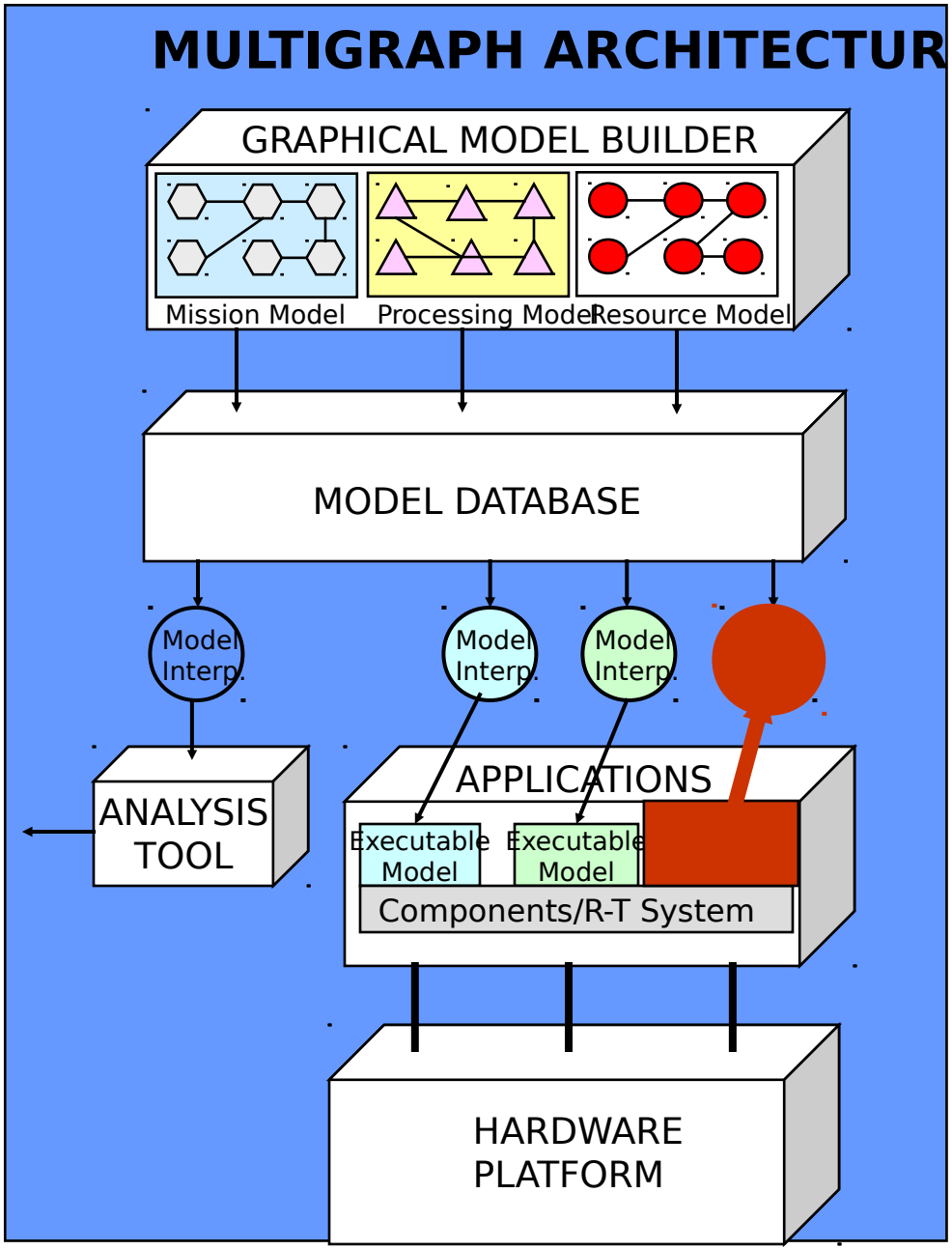
ADAPTATION/EVOLUTION IN THE BASELINE MGA:

- ❑ **MODEL-DRIVEN EVOLUTION**
- ❑ APPLICATION ADAPTATION



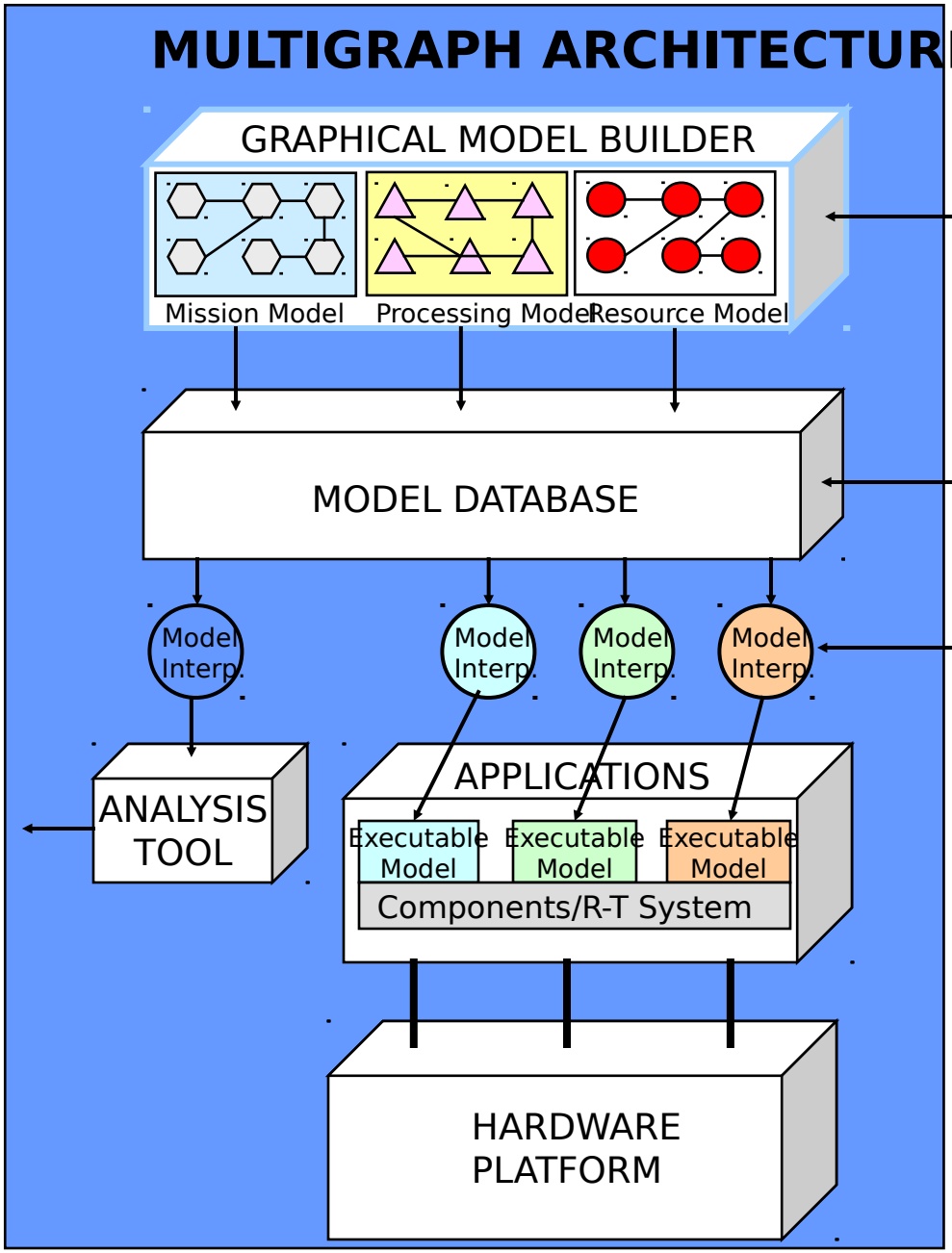
ADAPTATION/EVOLUTION IN THE BASELINE MGA:

- ❑ **MODEL-DRIVEN EVOLUTION**
- ❑ APPLICATION ADAPTATION



ADAPTATION/EVOLUTION IN THE BASELINE MGA:

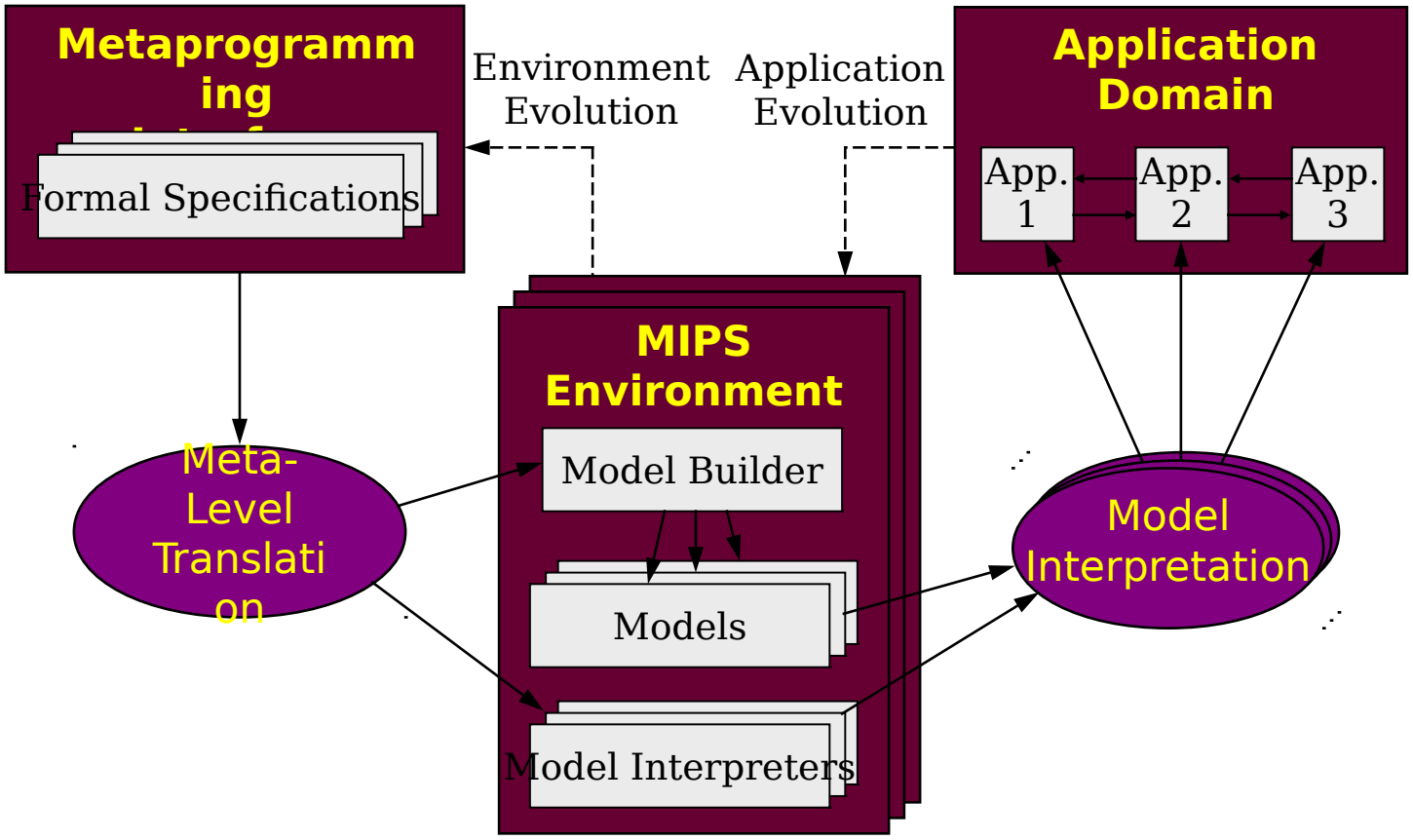
- ❑ MODEL EVOLUTION
- ❑ **ADAPTATION APPLICATION**



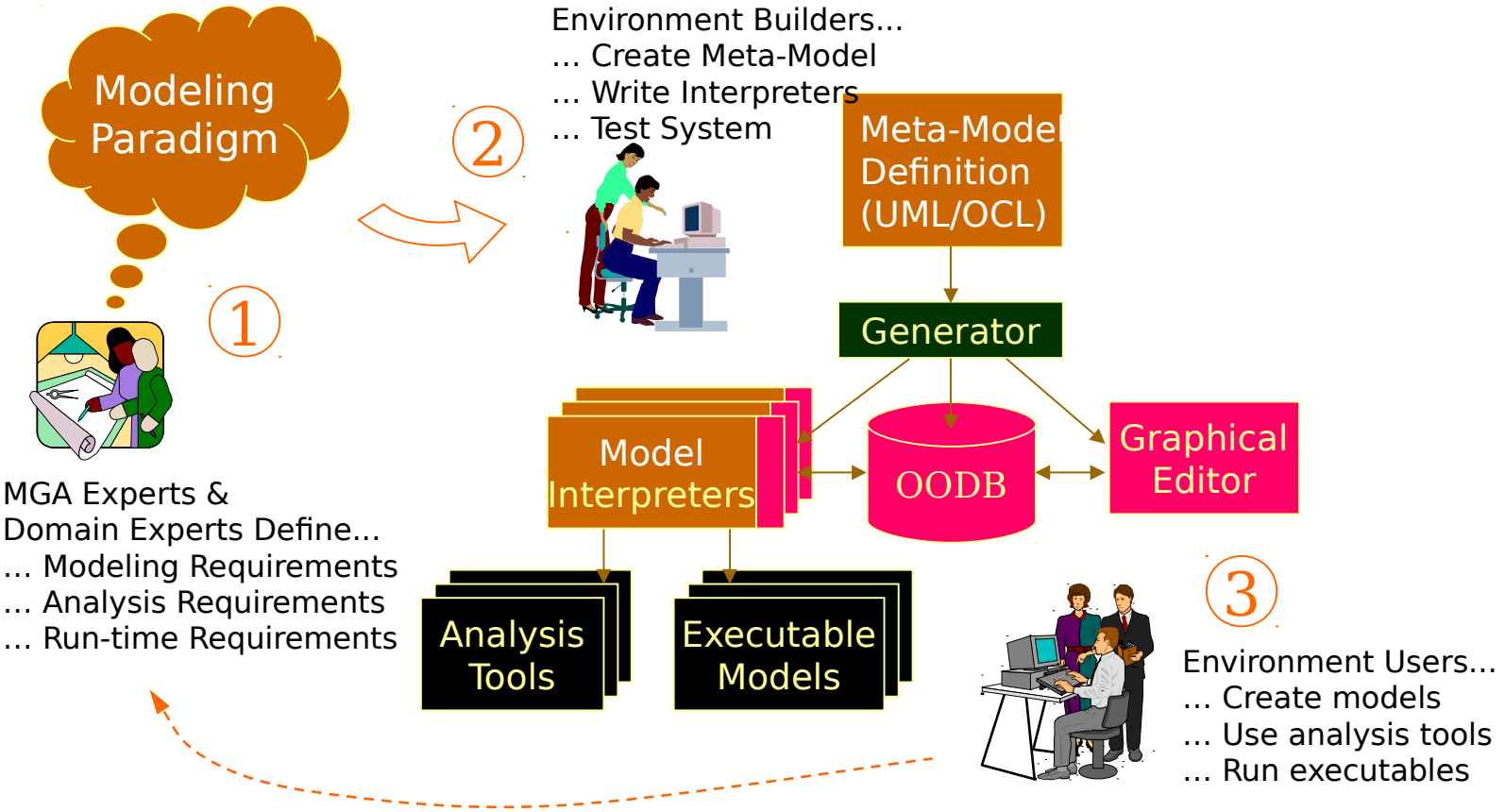
OBJECTIVES:

- **MGA Meta-level tools**
- **Environment Evolution**

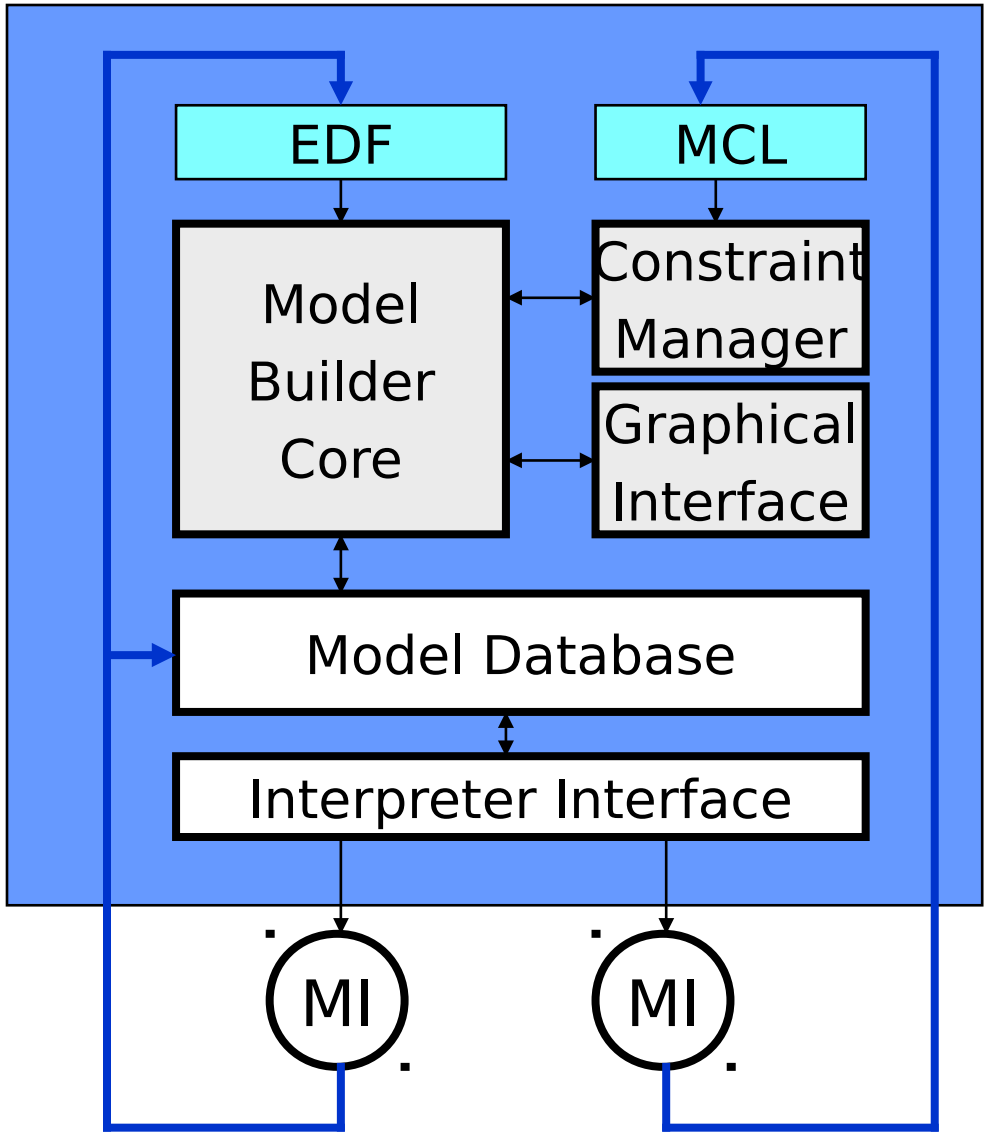
Model-Integrated Computing



Creating an Environment using the MultiGraph (MGA) Tools



Metaprogrammable Model Builder

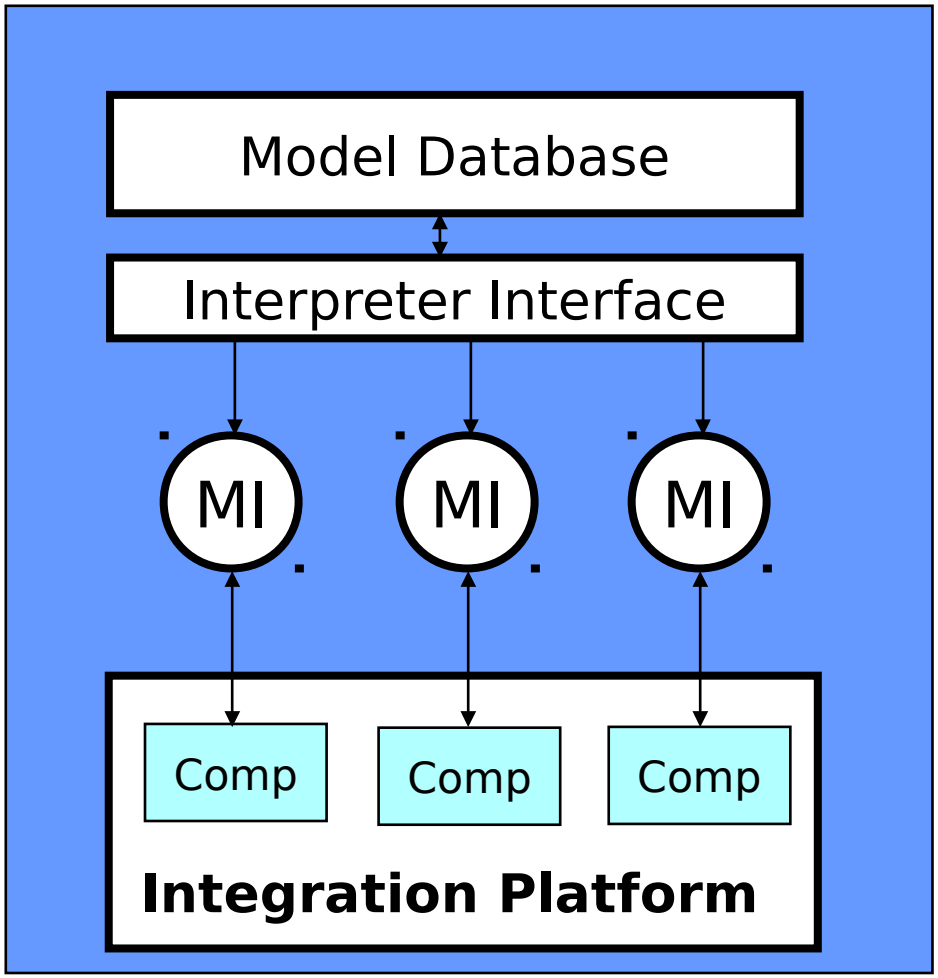


Meta-circular representati

1. Meta-meta model (UML/OC
2. Meta-model (UML/OCL)
3. Domain models
4. Data

MIC Systems ’98:

(active composition)



Object DB
OLE-CS..
MS Repository

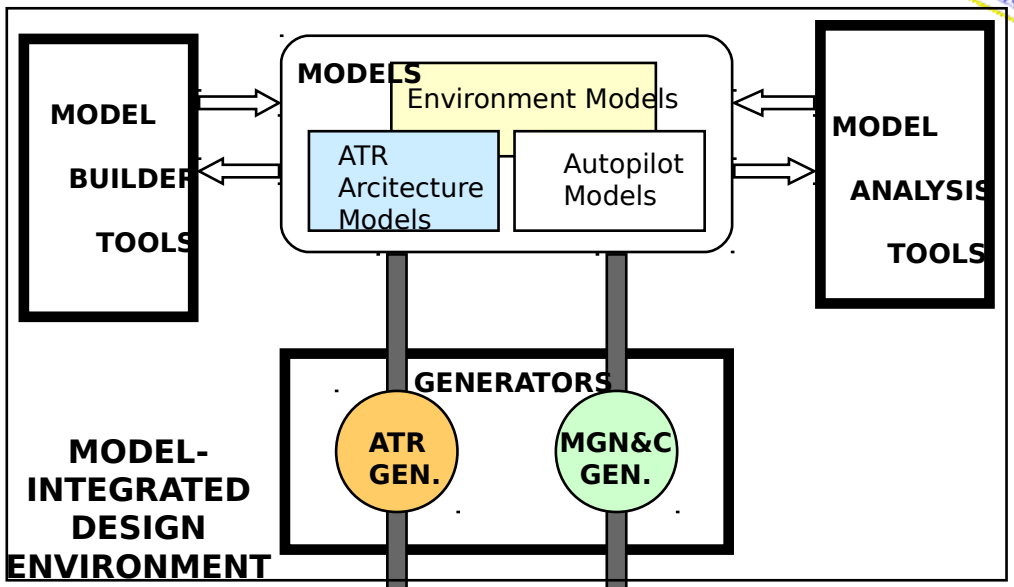
COM; *CORBA*

C++; JAVA; VB
MI Generator

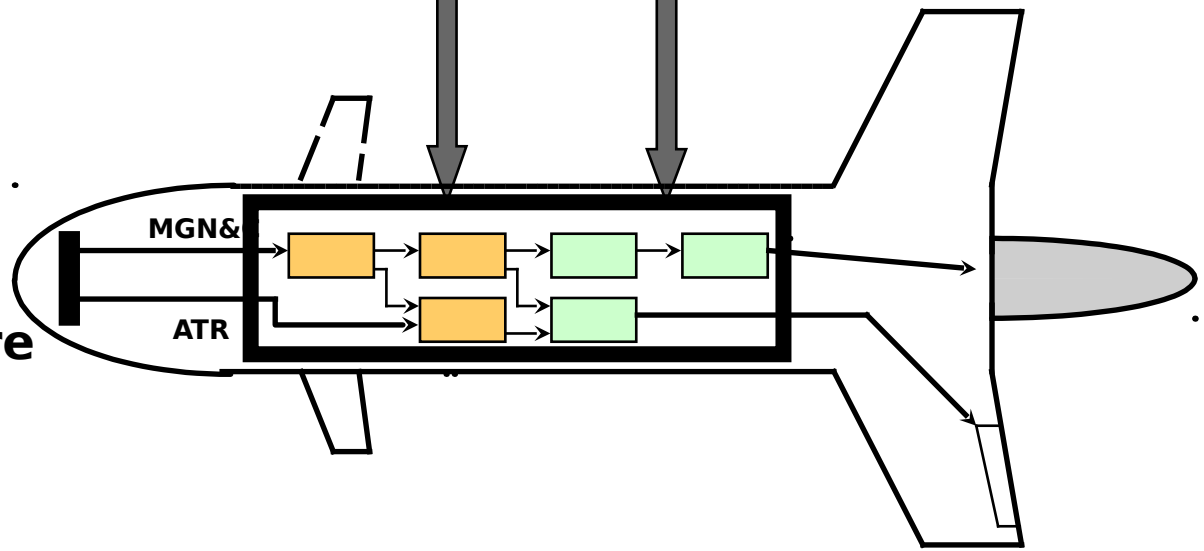
MCM (dataflow)
API-based
CORBA, DCOM

Adaptive Computing Systems:

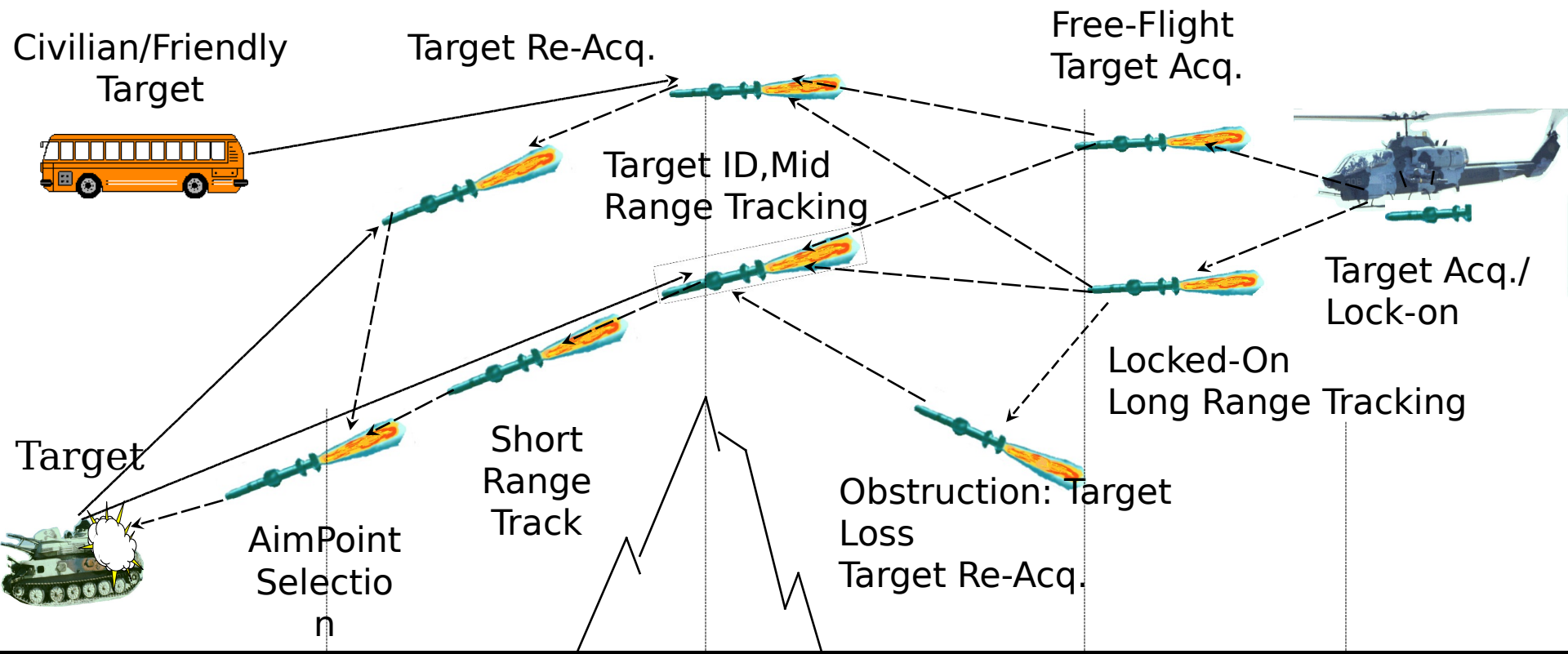
ENGINEERING PLATFORM

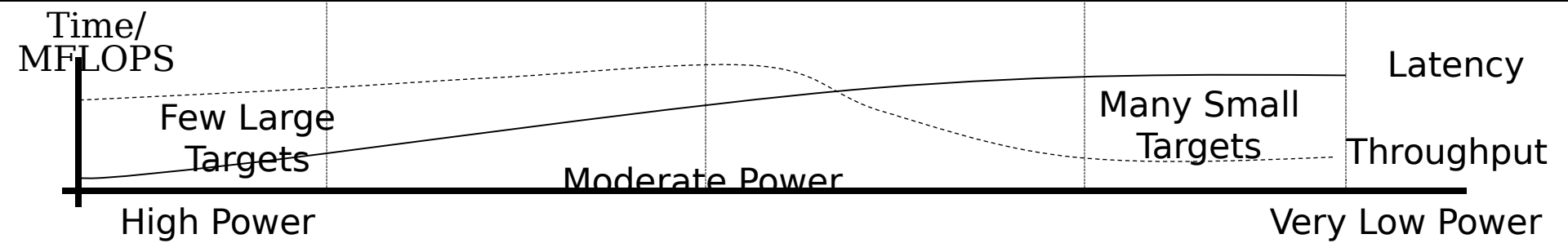


OPERATIONAL PLATFORM:
reconfigurable hardware architecture

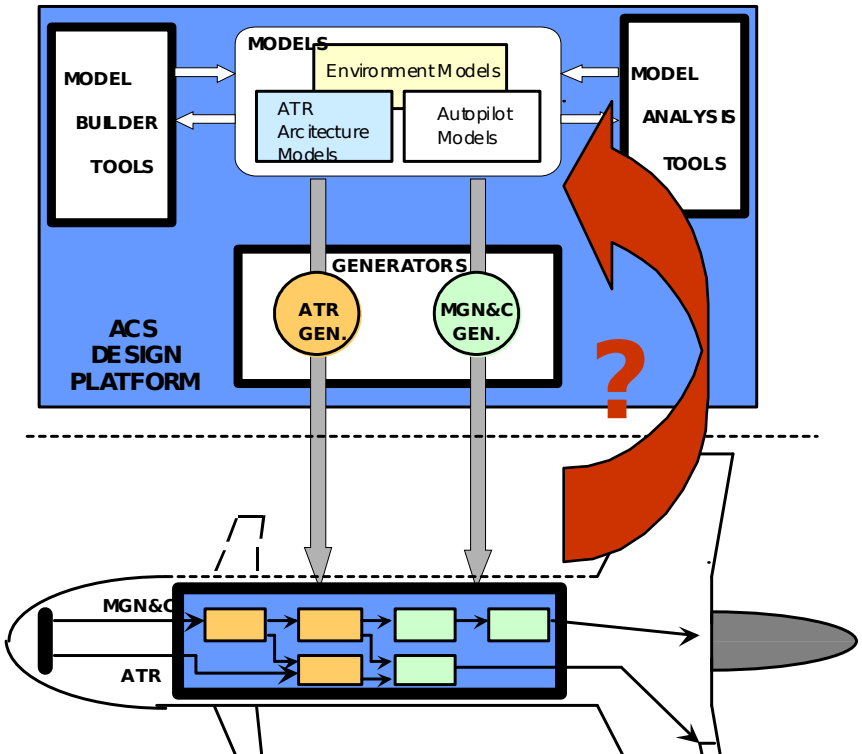


Adaptive ATR Application Scenario

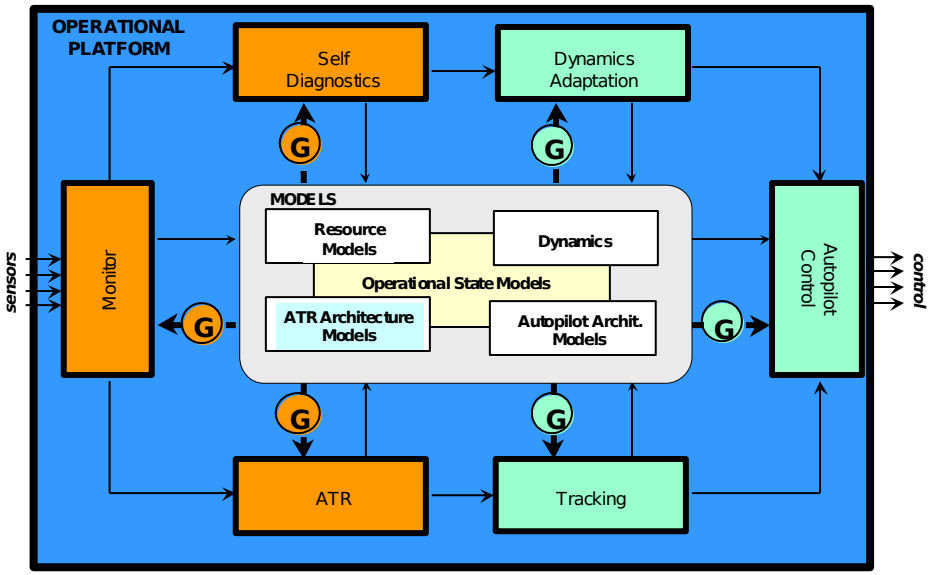




Reconfigurable Computing Platform:



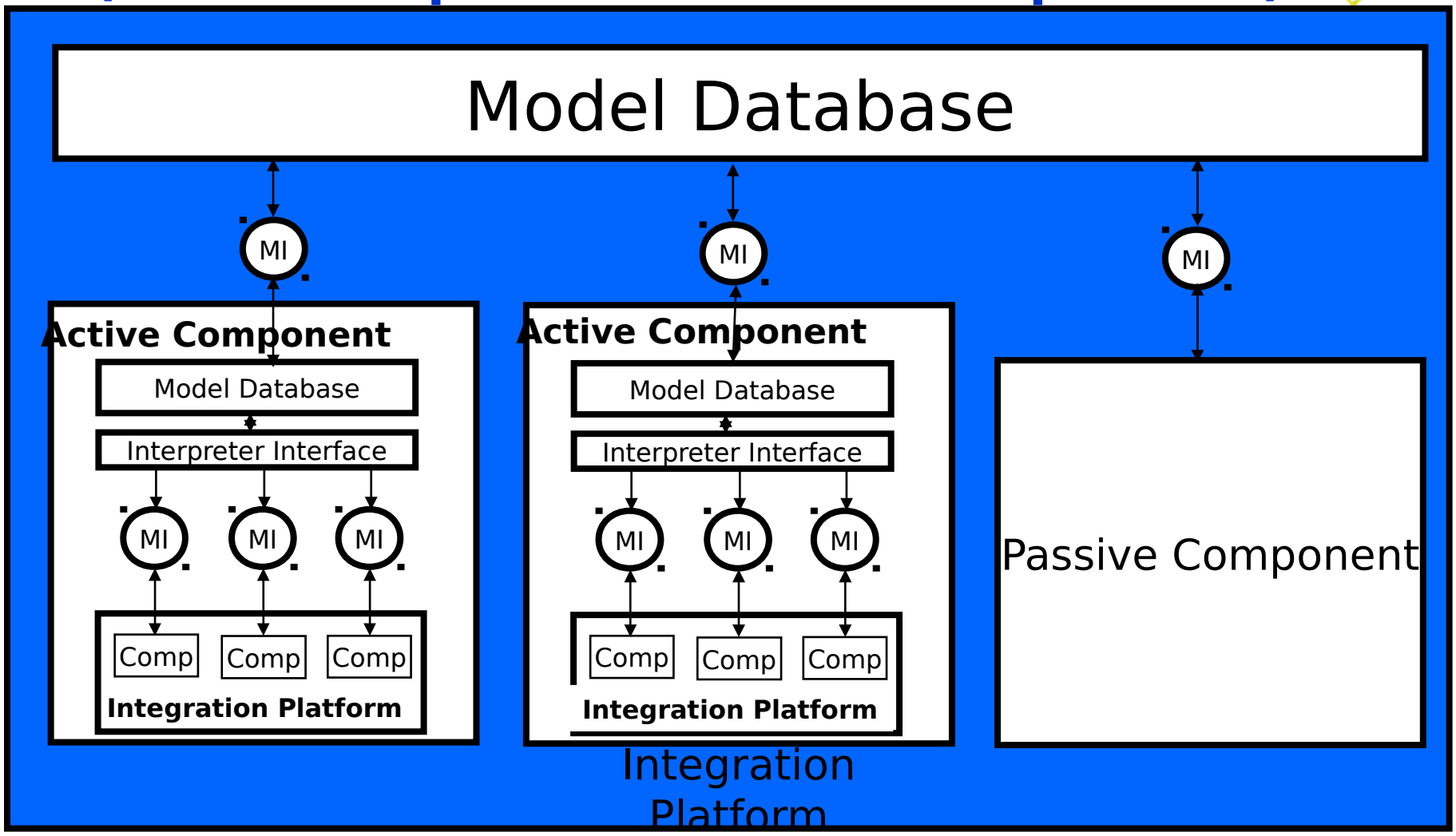
Active Software Compositi



- How to embed models and model interpre (generators) in applications?
- How to shift emphasis from verification of correctness to proof of "structural stabili
- How to utilize the technology to create a generation of autonomous, self-correcting systems?

MIC Systems ’2000:

(active components - active composition)



Selected Applications

- Saturn Site Production Flow System
- Integrated Test Information System (ITIS), USAF-AEDC
- International Space Station FDIR Modeling and Diagnosability Analysis System, Boeing/NASA
- Common Diagnostic Testbed for JSF, Boeing
- Computer Assisted Dynamic Data Monitoring and Analysis System, USAF-AEDC/NASA
- Activity Modeling Tool, DuPont
- Integrated Safety and Reliability Analysis Toolset, Sandia
- Embedded, adaptive missile ATR, DARPA
- Integration Environment for Real-Time, Distributed Flight Simulation (ARINC)

Saturn Site Production Flow

Saturn Site Production Flow
Production Flow

- Flow of raw materials, parts, sub-assembly through the operations
- A complex network of operations and buffers
- Production flow very critical to throughput

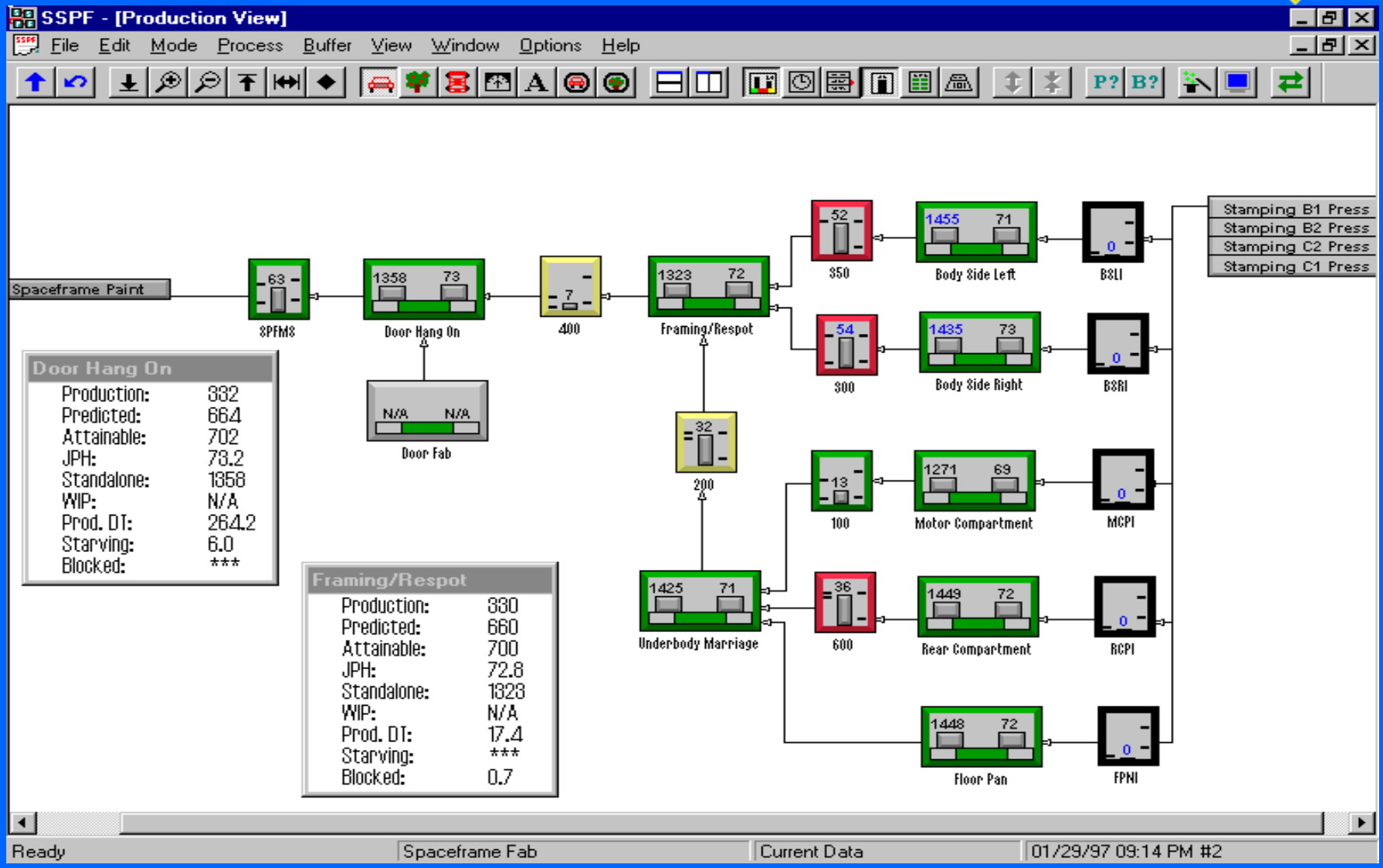
Saturn Site Production Flow
Data Collection and Storage

- Real-time collection
- Nearly 80,000 points
- Large-scale, distributed system
- Storage of historical data
- Hundreds of distributed clients
  - » LAN and WAN
  - » Real-time and historical data

Saturn Site Production Flow Analysis and Visualization

- Flexible access to real time/historical data
- Visualization
- Bottleneck identification
- What-If analysis
- Data tranding
- Interoperation with COTS
- Reports/charts generation

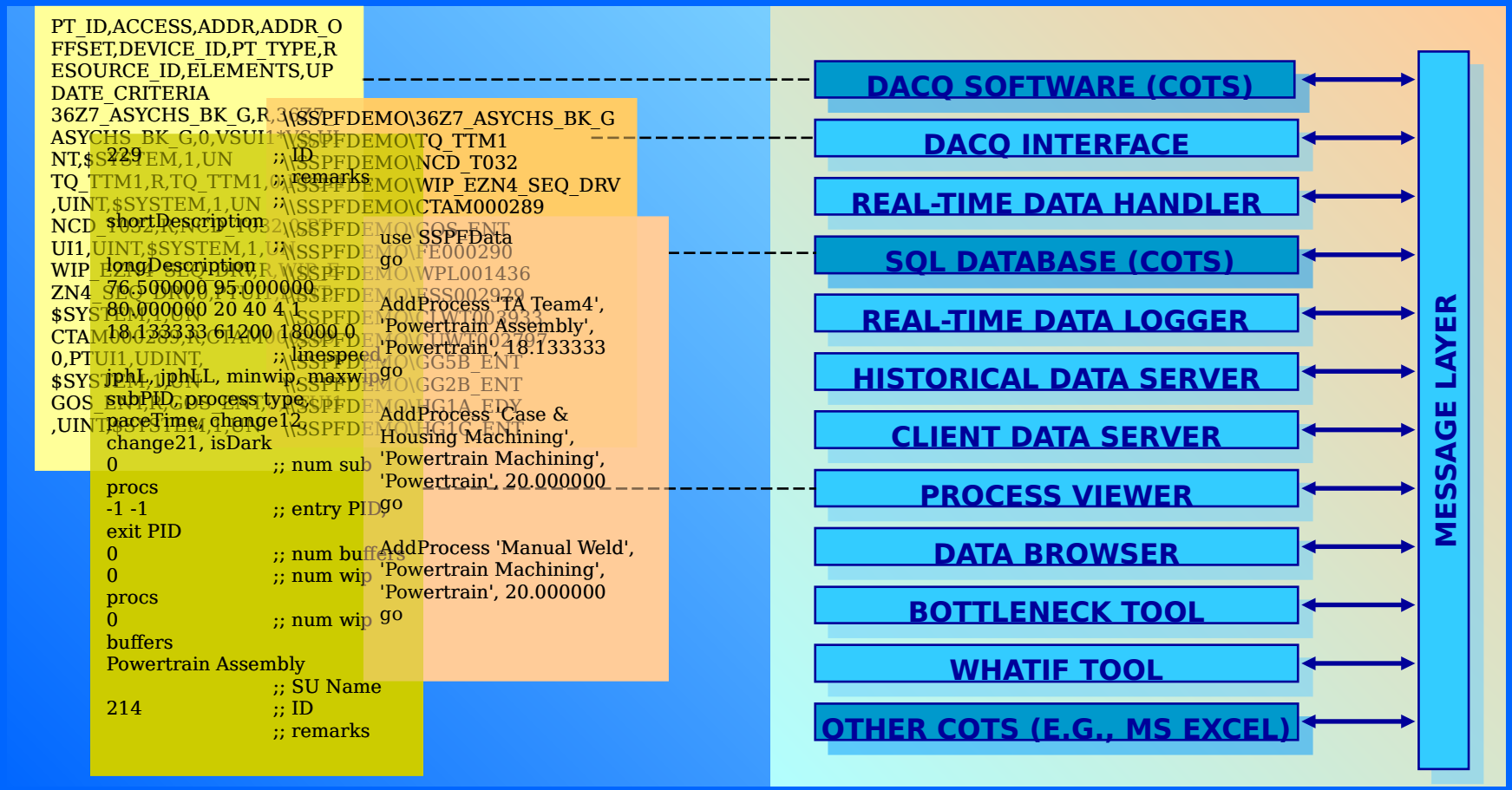
Real Time Viewer



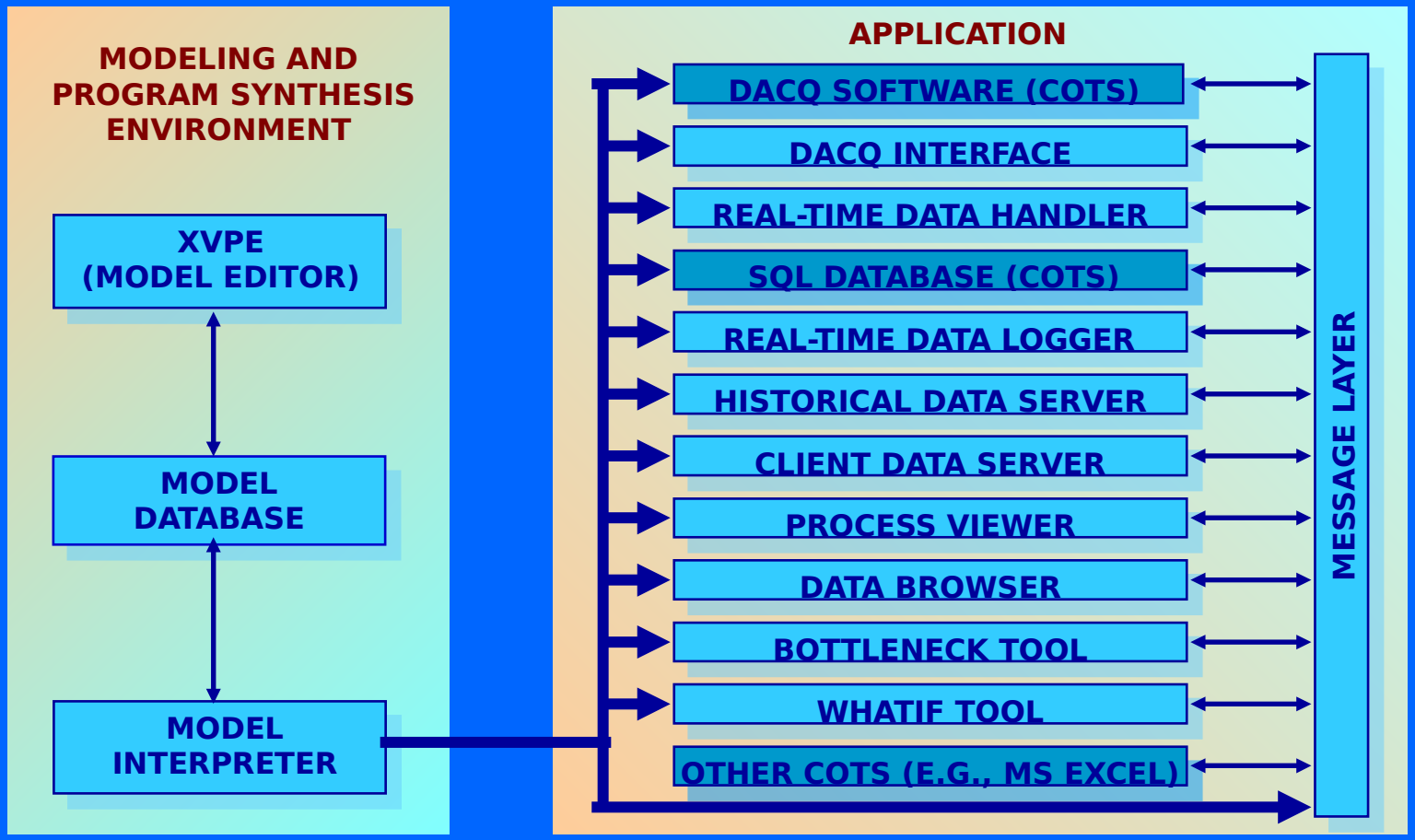
Saturn Site Production Flow
Software Engineering Issues

- Large-scale, complex, heterogeneous and continually changing environment
- Distributed, real-time application
- Changing requirements
- Integration with the plant
- Integration of diverse software sub-systems
  - » Custom and COTS
- Maintenance
- Evolution

Saturn Site Production Flow
"Tables"



Saturn Site Production Flow
Components



New Capabilities, Benefits

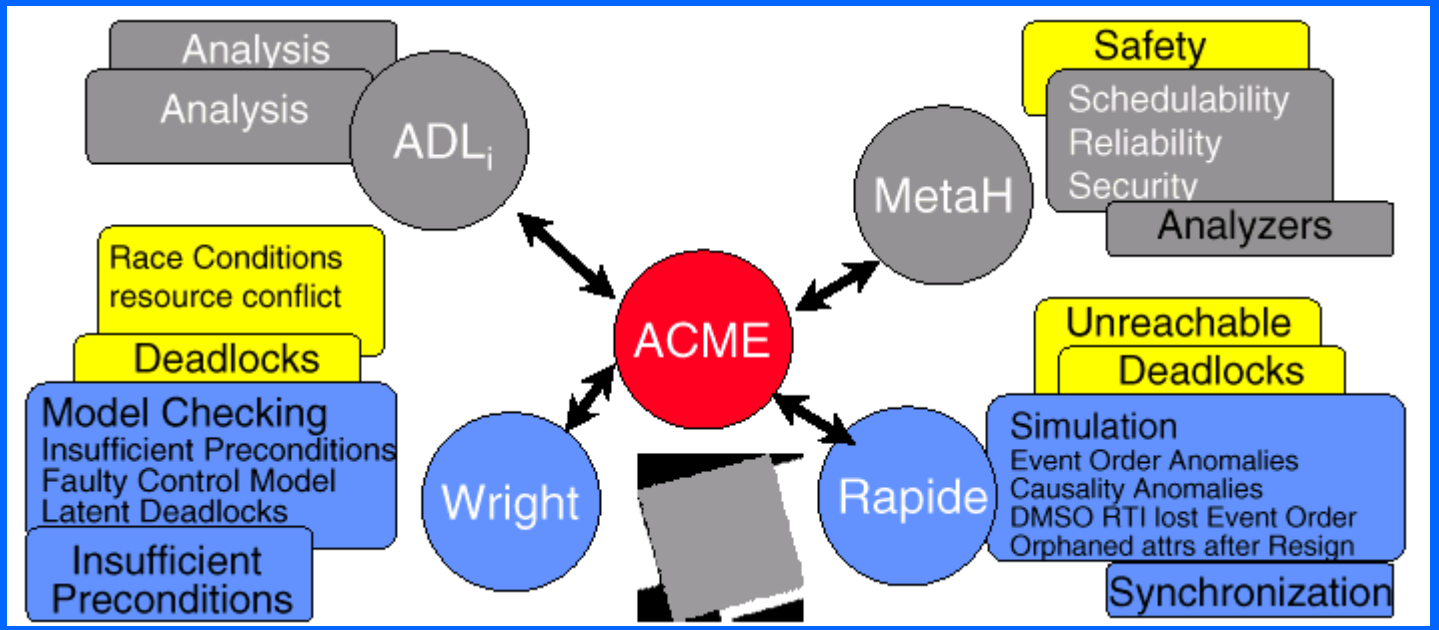
- 10-100 X decrease in system integration cost
- Enables evolution and adaptation of complex software
- Provides “true” end-user programmability

Other EDCS Technologies

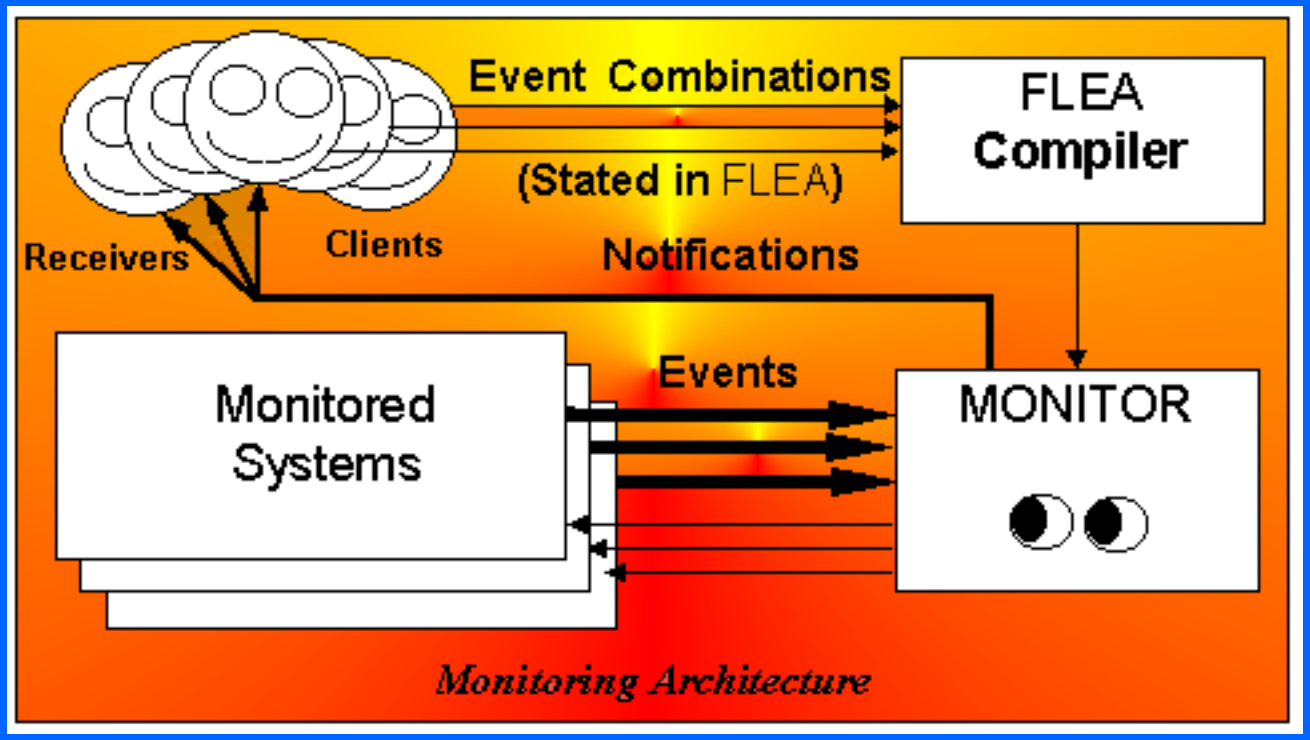
- Architectural Tools
- Monitoring Tool
- Configuration Management

Architectural Analysis

ACME developed as a common interchange mechanism
- Supports common static analysis services
- Provides tools access through ADL translation
- Facilitates development of domain-specific notations



Server (SoMoS)



* A generic, application-independent monitoring service
* Able to describe many useful event patterns
* Able to change criteria “on-the-fly” without massive code re-write o
* Complete event histories are maintained

Software Release Manager (SRM)

Description:

- SRM is used to manage the dependencies between the (often) distributed components of a complex software system, and automates the fielding of those components to a user site in order to ensure that a properly configured and operational system is installed

Benefits:

- Ease of Installation
- Better Guarantee of Interoperability Between Components During Upgrade
- Centralized Software Release for Distributed Components
- Dependency Information Capture and Evolution
- Automated Configuration Mgt

Problems being Addressed:

- Large System of Systems
- Multiple Versions
- Distributed Software
- Multiple Developers
- Site Specific Install
- COTS Software Upgrades
- Complex Dependencies
  - SW/HW Interfaces
  - Data & Control Flow

Demonstration Context:

- Theater Battle Management Core System (TBMCS) Software Release Management
  - Notional Demonstration with Realistic TBMCS Unclassified Data
  - Suggests a Way to Manage the Deployment of TBMCS Software Upgrades to Force and Unit Level Sites
  - Illustrates the Complexity of